OPTIMAL ASSIGNMENT OF TASK MODULES WITH PRECEDENCE FOR DISTRIBUTED PROCESSING BY GRAPH MATCHING AND STATE-SPACE SEARCH

LING-LING WANG and WEN-HSIANG TSAI


Index Terms: Task assignment, distributed processing, graph matching, topological tree, minimax criterion, task turnaround time, state-space search, A* algorithm.

1. Introduction.

A distributed computing system consists of two or more arbitrarily and incompletely interconnected processors. A concern in the research of distributed processing is how to utilize the processors evenly in a distributed computing system. This is the so-called task assignment problem [1, 9]. In such a problem, a task consisting of several modules is to be solved on a set of processors with the aim to reduce task turnaround time and to increase system throughput.

The purpose of reducing task turnaround time and increasing system throughput can be achieved by maximizing (or balancing) the utilization of resources and minimizing the communication between processors [2]. While minimizing interprocessor communication tends to assign the whole task to a single processor, load balancing tries to distribute the task modules evenly among the
processors. Therefore, there exists a conflict between these two criteria and a compromise must be made to find an optimum policy for task assignment.

Several approaches [1-16] have been suggested, including the graph-theoretic approach, the integer 0–1 programming approach, the heuristic approach, the simulated annealing method, and approaches used in the construction of parallel computing elements in VLSI. In most of the approaches, module precedence is neglected. Some have considered this problem but only allow unit-length modules in a task [10]. In this paper, we remove these constraints and consider optimal assignment of task modules with varying lengths and precedence relationship.

Each graph match corresponds to a task assignment. Cost values are defined in terms of a single unit, time, and a state-space search method [19] is used for finding the minimal-cost graph match corresponding to the optimal task assignment.

Inclusion of module precedence into the optimal solution is made possible by the use of topological module orderings constructed from the precedence relationship. The proposed model allows most constraints encountered in practice to be easily incorporated. On the other hand, the proposed approach guarantees to find an optimal solution. This is especially important for those applications where the resulting assignment will be run on a distributed system repeatedly.

In the remainder of this paper, we describe the system assumptions in Sec. 2, and the graph matching model in Sec. 3. In Sec. 4, we describe the minimax criterion and in Sec. 5, we derive the cost function, namely, the task turnaround time. In Sec. 6, the state-space search method is reviewed and applied to finding the optimal solution. Some illustrative examples are given in Sec. 7. Conclusions appear in the last section.

2. System assumptions.

Various assumptions made of the task and the distributed computing system are described in the following.

(1) The processors in the system are heterogeneous, and the modules may have different degrees of preference of the processors.

(2) Nonidentical communication links are used by the processors for message transmission, and module communication processes may have different degrees of preference of the links.

(3) The link between any two processors is symmetric. Therefore, the time to transmit messages from one processor to another is identical to that to transmit the same messages in the reverse direction.

(4) There exists a precedence relationship among the task modules. This means that if module \( m_1 \) is a predecessor of module \( m_2 \), \( m_2 \) cannot be executed before